Year 7 Curriculum Intent

The Year 7 Computer Science curriculum at The Kingsway School encourages students to explore the fundamental concepts of Computational Thinking, Problem Solving, Programming, Computer Systems, Digital Artefacts and Digital Literacy in a subject domain which encompasses real world links to how computing is embedded into modern day life. The curriculum aims to build directly on the knowledge from Key Stage 2 and is designed to stretch and challenge students' ability within Computer Science. Students will explore both a practical and theory-based approach, and aims to encourage resilience in solving problems with working solutions using PRIMM (Predict, Run, Investigate, Modify, Make)

The Year 7 focus is on preparing students with the essential digital literacy and digital life skills as well as laying the groundwork with essential computing concepts required for understanding how computers work and paving the way for more advanced programming and problem-solving in future years.

Essential Knowledge Summary

Schemata 1: Introduction to CS at TKS and Safety

Composite Knowledge

The aim of this unit is to provide an introduction and refresher on computer use, focussing on the school systems, and file management as well as essential awareness of online safety and security. This unit will prepare students for their future learning at The Kingsway School, within Computer Science and other subjects by developing their digital literacy skills. *Some students have very limited exposure to general personal computer use and may only have basic digital literacy skills related to smartphone/tablet use.*

Component Knowledge:

Foundational Knowledge:

Declarative Knowledge:

- The Google Systems used by the school.
- Individual One-Drive.
- Essential digital literacy skills.
- Essential keyboard shortcuts for Screenprint/Copy/Paste.
- File organisation/management.
- Social media and Fake News issues.
- Online Security and staying safe online.

Procedural Knowledge:

- Demonstrate the basic use of a keyboard and a mouse.

Schemata 2: Office Applications and Spreadsheets

Composite Knowledge:

Students will learn to use the three main "office applications" used globally in business and education today, to produce well-presented professional documentation: namely word processing, presentation, and spreadsheet applications. Students will learn to use spreadsheet software to store, organise, format, and process information using formulas and functions for a specific purpose by creating a financial plan for an event. Abstracting a real-world situation into data forms a key part of much of the future programming work students will undertake. Managing, organising, processing and presenting data is also useful in a variety of other contexts. *Many students will not have used or seen a spreadsheet prior to this unit.*

Component Knowledge:

Foundational Knowledge:

Declarative Knowledge:

- Microsoft suite of office applications: Word, PowerPoint, Excel.
- Google suite of office applications: Docs, Slides, Sheets.
- Common application icons, buttons and menu items.
- Collecting, organising and entering data.
- Formatting spreadsheets.
- Spreadsheet cell references.
- Spreadsheet formulas.
- Spreadsheet functions.

Schemata 3: Computational Thinking / Introduction to Scratch

Composite Knowledge:

This unit lays the groundwork for computer programming by introducing students to the fundamental concepts of computational thinking, problem decomposition, algorithms and automation. Students are then introduced to sequenced programming in a block-based language and will use basic commands and structures to solve a series of small challenges, but without the burden of needing the knowledge of text-based syntax and command words. *Some students will already be familiar with some basic block-based programming and will progress quickly through the initial work and will have the chance to develop and show their additional skills in the features they program into their game*.

Component Knowledge:

Foundational Knowledge:

Declarative Knowledge:

- Computational Thinking.
- Decomposition / abstraction / algorithms.
- Events in Scratch.
- Actions in Scratch.
- Programming Constructs – sequence, selection and iteration.
- Mathematical and Logical Operators.

Procedural Knowledge:

- Understand that algorithms are a set of step-by-step instructions.

● Successfully log into the school computer network. ● Identify and log into the appropriate applications needed under different educational circumstances. ● Repeatedly use essential keyboard shortcuts. ● Understand and demonstrate the need for appropriate file organisation and management. ● Recognise the security and safety issues associated with specific social media platforms. ● Know how to spot online scams and phishing attacks. ● Understand how to spot fake news online. **Upper Hierarchical Knowledge** ● Explain how to use social media platforms safely and where to get help. ● Based on specific criteria, articulate their own opinions on the validity of online news stories to determine whether it is true or not.	● Spreadsheet graphs. **Procedural Knowledge:** ● Create, edit and save Microsoft “office” documents. ● Create, edit and save Google “office” documents. ● Use different formats of text in all documents, including size, colour and font. ● Explore and use bullets, shapes, tables and pictures in word processing and presentation documents. ● Explore and use layout design, animations and transitions in presentation documents. ● Use spreadsheet formulas to add, subtract, multiply, and divide the contents of two spreadsheet cells. ● Use spreadsheet functions to calculate the sum of a range of numbers. ● Use spreadsheet functions to calculate the maximum, minimum and average numbers in a range of numbers. ● Use the spreadsheet fill down method to efficiently manage data and formulas. ● Use graphs to present spreadsheet data. **Upper Hierarchical Knowledge** ● Use problem-solving skills to break down a specified financial planning problem and implement a solution using spreadsheet formulas and functions. ● Model and articulate the impact on a financial budget by adjusting the data in a financial plan using a spreadsheet.	● Recognise and use Events and Actions in Scratch. ● Program simple routines in Scratch using sequence. ● Program simple routines in Scratch using selection. ● Program simple routines in Scratch using iteration. ● Understand and use sequence, selection and iteration together to automate the movement of a character/ sprite around the screen, collecting and eating apples. **Upper Hierarchical Knowledge** ● To be able to add additional features to the apple collecting game such as using variables to keep track of the score and/or the time taken to collect all the apples. ● Develop debugging skills by finding and fixing errors (bugs) in a program.
Schemata 4: Scratch Development – Project	**Schemata 5: Data Representation 1**	**Schemata 6: Physical Programming**
Composite Knowledge: Building on the skills learnt in the previous unit, and using the same block-based programming language, students will bring those skills together to develop a complete Pac-Man game. Students will need to simplify the complex game elements into manageable representations for programming, ensuring their game is	**Composite Knowledge:** Students begin to learn how binary digits are used to represent different types of data, more specifically numbers, text, images and colour. By exploring the fundamental concept of binary digits and their role in computing, students will gain a foundational understanding of how computers	**Composite Knowledge:** This unit bridges the gap between abstract ideas and concrete creations through the hands-on programming of digital devices which interact with the physical world, where students can see the direct impact of their code. Using the BBC micro:bit and the online MakeCode editor students will be able to transform abstract coding concepts into tangible, interactive projects,

playable and which follows the rules and objectives of the well-known Pac-Man game. The game must include multiple characters and multiple goals/objectives, and the main Pac-Man character must respond to human input via the keyboard. **Component Knowledge:** **Foundational Knowledge:** Declarative Knowledge: ● Game play, including its objective, rules and character interactions. ● Simplify complex game elements. ● Implement logical and ordered execution of instructions in a program. ● Enable automated decisions based upon conditions present. ● Implement instructions which will be repeated multiple times. **Procedural Knowledge:** ● Explain how to play the Pac-Man game and what the goal of the game is. ● Identify the multiple characters in Pac-Man and their individual roles. ● Describe the maze and how it affects the game. ● Break down the game into smaller parts (e.g., Pac-Man, goals, enemies) **Upper Hierarchical Knowledge** ● Recognise and describe how patterns exist in the way characters move or how points are scored. ● Understand that game elements like characters are simplified versions of real-world things. ● Develop debugging skills by finding and fixing errors (bugs) in a program.	operate, store data, and communicate, allowing a deeper understanding to be pursued later in the curriculum. **Component Knowledge:** **Foundational Knowledge:** Declarative Knowledge: ● Binary to denary conversions. ● Denary to binary conversions. ● The ASCII Character Set. ● Digital pictures pixels, resolution and colour depth. ● Factors affecting digital picture quality and file size. ● 24-bit pictures. **Procedural Knowledge:** ● Carry out binary to denary and denary to binary number conversions. ● Understand the evolution of Character Sets. ● Decode and encode binary digits using the ASCII Character Set. ● Create and encode a picture within the limitations set on pixels, resolution and colour depth. ● Interpret and recreate a picture based on given binary digits, pixels, resolution and colour depth. ● Explain the relationship between picture quality and file size with resolution and colour depth. **Upper Hierarchical Knowledge** ● Predict the file size of digital pictures given specific meta-data. ● Explain why sending and downloading high quality digital photographs and uploading and streaming high quality digital videos is resource intensive.	gaining a more intuitive and practical understanding of coding principles, as well as gaining practical experience and understanding of inputs, outputs, and variables in real-world contexts. **Component Knowledge:** **Foundational Knowledge:** Declarative Knowledge: ● Real-world buttons, sensors, and LEDs. ● Algorithms as a set of instructions. ● Movement detection via the accelerometer. ● Sequence, selection, iteration and randomisation. ● Use of variables to track data. ● Input and output of data. **Procedural Knowledge:** ● Understand the micro:bit is a tiny computer which needs instructions in code to make it work. ● Create instructions in code that the micro:bit can understand and then transfer them to the micro:bit. ● Code the micro:bit to show simple animations on its LED display output. ● Code the micro:bit to make different outputs happen depending on different inputs. ● Code the creation of random numbers in a fixed range. **Upper Hierarchical Knowledge** ● Describe why sequence and timing is important when making an animation. ● Understand that animations create an illusion of movement by showing a sequence of still images. ● Understand that inputs and outputs involve the flow of data in and out of computers. ● Evaluate the fairness of computer simulations.
Year 7 Final Composite Knowledge End Point		

By the end of year 7 the Computer Science department aims to ensure all students can safely, securely, respectfully and responsibly access the school computer network and understand how to protect their identity and privacy online. They will know how to recognise inappropriate content, contact and conduct and know how to report concerns. They will know how to successfully design, create and edit digital artefacts using “office” applications, and block-based programming languages. They will have undertaken creative projects to achieve challenging goals, including collecting and analysing data, as well as solving a variety of computational problems through programming activities. They will also have gained a fundamental understanding of data representation through binary numbers, as well as touching on the concept of data input and output, in readiness for their year 8 studies.

Year 8 Curriculum Intent

The Year 8 Computer Science curriculum at The Kingsway School encourages students to explore the fundamental concepts of Computational Thinking, Problem Solving, Programming, Computer Systems, Digital Artefacts and Digital Literacy in a subject domain which encompasses real world links to how computing is embedded into modern day life. The curriculum aims to build directly on the knowledge from Year 7 and is designed to stretch and challenge students' ability within Computer Science. Students will explore both a practical and theory-based approach, and aims to encourage resilience in solving problems with working solutions using PRIMM (Predict, Run, Investigate, Modify, Make)

The Year 8 focus is on providing students with an understanding of how computer systems and computer networks work as well as expanding on their year 7 learning to gain a deeper understanding of how data is represented in binary form. The focus then turns to techniques for efficient and logical problem-solving as well as becoming familiar with more advanced programming using a text-based programming language.

Year 8 Essential Knowledge Summary

Schemata 1: Computers Systems and Hardware

Composite Knowledge:

This unit reveals the inner workings of computer systems, uncovering the relationship between the hardware (physical components) and the software (the instructions). Students will learn how software is executed by the central processing unit (CPU), stored in memory (RAM), and how it interacts with the humans via the input and output peripheral devices. By understanding this interaction, students will gain a deeper appreciation of how computer technology works.

Component Knowledge:

Foundational Knowledge:

Declarative Knowledge:

- General purpose computers and embedded systems.
- Input, Process, Output.
- Computer hardware.
- The CPU and the FDE Cycle.
- Software and the Operating System.

Procedural Knowledge:

- Describe the characteristics of the different types of computers.
- Explain the difference between a general-purpose computer and an embedded system.

Schemata 2: Networks / Internet / WWW

Composite Knowledge:

This unit begins by defining a computer network and addressing the benefits of networking, before covering how data is transmitted across networks using protocols. The types of network hardware required are explained, as are the differences between wired and wireless data transmission. Learners will develop an understanding of the terms 'The Internet' and 'The World Wide Web', and of the key services and protocols used.

Component Knowledge:

Foundational Knowledge

Declarative Knowledge:

- Computer networks.
- Protocols, packets and addressing.
- Networking hardware.
- Wired and wireless networks.
- The Internet and internet services.
- The World Wide Web.

Procedural Knowledge:

- Define what a computer network is and explain how data is transmitted between computers across networks.
- Define 'protocol' and provide examples of non-networking protocols.
- List examples of the hardware necessary for connecting devices to networks.

Schemata 3: Data Representation 2 and Logic

Composite Knowledge:

Building on the Data Representation unit in Year 7, students will continue to learn how binary digits are used to represent different types of data, more specifically sound. Hexadecimal numbers and how they are used and converted will also be covered. Students will also carry out mathematical operations using binary numbers, including binary addition and binary shifts. Students will then look at Boolean logic and logic gates to gain insight into how computer electronic circuits perform basic logical operations and decision making.

Component Knowledge:

Foundational Knowledge:

Declarative Knowledge:

- Binary additions.
- Binary shifts – left and right.
- Converting Analog sound to Digital sound.
- Frequency and amplitude.
- Sample rate and sample resolution.
- Factors affecting digital sound quality and file size.
- Hexadecimal to denary conversions.
- Denary to hexadecimal conversions.
- Boolean logic, logic gates and logic circuits.

Procedural Knowledge:

- Solve binary additions using the four rules of binary additions.
- Validate the solved binary additions by converting back to denary.

● Differentiate between input and output peripheral devices. ● Describe the functions of the different types of computer hardware. ● Explain how the CPU works. ● Describe how the CPU interacts with the user and the other hardware devices. ● Describe the function of the Operating System. **Upper Hierarchical Knowledge** ● Analyse how the hardware components used in computing systems work together in order to execute programs.	● Compare wired to wireless connections and list examples of specific technologies currently used to implement such connections. ● Define 'bandwidth', using the appropriate units for measuring the rate at which data is transmitted. ● Define what the internet is verses what the WWW is. ● Describe how services are provided over the internet. ● Describe components (servers, browsers, pages, HTTP and HTTPS protocols, etc.) and how they work together. **Upper Hierarchical Knowledge** ● Explain the term 'connectivity' as the capacity for connected devices ('Internet of Things') to collect and share information about me with or without my knowledge (including microphones, cameras, and geolocation). ● Describe how internet-connected devices can affect me.	● Solve binary shifts, both left and right. ● Validate the solved binary shifts by converting back to denary. ● Encode a sound wave within the limitations set on sample rate and sample resolution. ● Explain the relationship between sound quality and file size with sample rate and sample resolution. ● Describe the NOT, AND, and OR logical operators, and how they are used to form logical expressions. ● Use logic gates to construct logic circuits and associate these with logical operators and expressions. **Upper Hierarchical Knowledge** ● Predict the file size of digital sound recordings given specific meta-data. ● Explain why uploading and streaming high quality sound recordings is resource intensive. ● Describe how hardware is built out of increasingly complex logic circuits, and since hardware is built out of logic circuits, data and instructions alike need to be represented using binary digits.
Schemata 4: Computational Thinking / Algorithms	**Schemata 5: Python Programming 1 - Turtle**	**Schemata 6: Python Programming 2**
Composite Knowledge: Before computers can be used to solve a problem, the problem itself and the ways in which it could be resolved must be understood. Computational thinking techniques help with these tasks. Students will learn how to break down complex problems into smaller, more manageable parts, recognise patterns, abstract unnecessary details, develop step-by-step solutions (algorithms), and create simplified models of real-world situations. The aim of this unit is to foster a problem-solving mindset and provide students with the essential tools for tackling challenges both logically and systematically. **Component Knowledge:** **Foundational Knowledge:**	**Composite Knowledge:** This unit builds upon the block-based programming in year 7, and draws upon the Computational Thinking skills learnt in the previous unit. It is an introduction to programming in a text-based language designed to make programming more approachable for beginners. It starts by introducing a virtual turtle robot, leading to the use of variables, loops, event handling and interactive graphics. Students will become familiar with textual programming statements whilst having fun producing coloured graphics, allowing a deeper understanding of a text-based language to be pursued later in the curriculum. **Component Knowledge:** **Foundational Knowledge:** Declarative Knowledge:	**Composite Knowledge:** Students begin to learn programming commands and structures for a text-based programming language in a real-world development environment. The learning in this unit starts with simple programs involving input and output, and gradually moves on through arithmetic operations, randomness, selection, and iteration. Students will adopt the five stages of PRIMM within their program development activities. **Component Knowledge:** **Foundational Knowledge:** ● Display messages on the screen. ● Capture keyboard input from the user. ● Assign values to variables. ● Data Types and Casting.

Declarative Knowledge: ● Computational Thinking Techniques ● Decomposition ● Pattern Recognition ● Abstraction ● Algorithms and Flow Charts ● Evaluating Solutions **Procedural Knowledge:** ● Break down simple tasks into smaller steps. ● Spot patterns in everyday things. ● Remove unnecessary details from a problem and focus on the important aspects. ● Write simple instructions for others to follow. ● Use computational thinking to solve challenges. ● Draw and interpret flow charts using flow chart symbols. **Upper Hierarchical Knowledge** ● Recognise scenarios where each of these computational thinking techniques is applied ● Describe the difference between algorithms and computer programs.	● Turtle Library. ● Moving a Turtle. ● Using For Loops. ● User-Input Shapes. ● Using Functions and Parameters. ● Event Handling. **Procedural Knowledge:** ● Build an understanding of what it means to express instructions in a formal language. ● Understand the basics of Python programming. ● Use basic Turtle commands forward(), backward(), left(), right() and pencolor(). ● Use loops in Python to simplify repetitive tasks. ● Draw complex geometric shapes using loops and Turtle commands. ● Add colour to Turtle drawings. ● Create and use functions to ensure code is organised efficiently. ● Create interactive graphics that respond to user inputs. **Upper Hierarchical Knowledge** ● To be able to create more complex and creative designs, such as colourful spirals or multi-shape patterns, by combining loops and colour changes effectively. ● To be able to design and implement a complex interactive project, like guiding the turtle through a maze or creating a drawing tool that responds to different inputs.	● Arithmetic Expressions. ● Relational Operators. ● If-else / if-elif-else statements. ● While Loops. ● Boolean Variables. **Procedural Knowledge:** ● Write simple Python programs that display messages, assign values to variables, and receive keyboard input. ● Locate and correct common syntax errors. ● Use simple arithmetic expressions in assignment statements to calculate values. ● Receive input from the keyboard and convert it to a numerical value. ● Use binary selection (if, else statements) to control the flow of program execution. ● Generate and use random integers. ● Use multi-branch selection (if, elif, else statements) to control the flow of program execution. ● Describe how iteration (while statements) controls the flow of program execution. ● Use variables as counters in iterative programs. **Upper Hierarchical Knowledge** ● Describe what algorithms and programs are and how they differ. ● Combine sequence, iteration and selection to control the flow of program execution.

Year 8 Final Composite Knowledge End Point

By the end of year 8 the Computer Science department aims to ensure all students have a fundamental understanding of how computer systems work, as well as how hardware components work together to follow the instructions of the software and operating systems. They will also know how computer networks enable communication between multiple computer systems and be able to explain the characteristic of and difference between the internet and the world wide web. They will have undertaken creative projects to achieve challenging goals, including solving a variety of computational problems through computational thinking and programming activities using a text-based programming language. They will also have gained a deeper understanding of data representation through binary numbers, as well as understanding simple Boolean logic and some of its uses in circuits and programming.

Year 9 Curriculum Intent

The Year 9 Computer Science curriculum at The Kingsway School encourages students to explore the fundamental concepts of Computational Thinking, Problem Solving, Programming, Computer Systems, Digital Artefacts and Digital Literacy in a subject domain which encompasses real world links to how computing is embedded into modern day life. The curriculum aims to build directly on the knowledge from Year 8 and is designed to stretch and challenge students' ability within Computer Science. Students will explore both a practical and theory-based approach, and aims to encourage resilience in solving problems with working solutions using PRIMM (Predict, Run, Investigate, Modify, Make)

The Year 9 focus is on empowering students to become informed and responsible digital citizens by deepening their understanding of cyber security and defensive strategies, as well as studying the impact technology is having on society, adding context for the computers' place in the world and further keeping students informed on the responsible use of technology. The focus then turns to reinforcing and building upon their year 8 text-based programming knowledge by using string methods, data structures, data validation and error handling. Finally, students will discover how to create 2D animations using the on-line tool Wick Editor and also create their own web page using HTML and CSS code.

Essential Knowledge Summary

Schemata 1: Cyber Security / Cryptography	Schemata 2: Evolution and Impact of Tech / AI	Schemata 3: Python Programming 3
Composite Knowledge: This unit takes the learners on an eye-opening journey of discovery about techniques used by cybercriminals to steal data, disrupt systems, and infiltrate networks. The learners will start by considering the value of their data to organisations and what they might use it for. They will then look at social engineering techniques used by cybercriminals to try to trick users into giving away their personal data. The unit will look at the more common cybercrimes such as hacking, DDoS attacks, and malware, as well as looking at methods to protect ourselves and our networks against these attacks. **Component Knowledge:** **Foundational Knowledge:** Declarative Knowledge: ● Data and Information. ● Data Protection Act. ● Social Engineering. ● Hacking. ● Malware. ● Impact of Cyber Threats and Managing Risk. ● Encryption / Cryptography. **Procedural Knowledge:** ● Understand the value of our data to criminals. ● Identify what happens to data entered online. ● Explain the need for the Data Protection Act.	**Composite Knowledge:** This unit follows the history of key technologies, such as the internet, WWW, mobile phones, satellites, robots and artificial intelligence. It looks at their growth and evolution, as well as the impact these technologies are having on the world today. With GCSE options in mind, this unit also introduces students to some of the individuals who work in each of these areas and looks at the roles they do, shining a spotlight on some of the careers available in these Computing and Technology Industries. **Component Knowledge:** **Foundational Knowledge:** Declarative Knowledge: ● Evolution of the Internet. ● Evolution of the World Wide Web. ● Evolution of Mobile Phones. ● Evolution of Satellites. ● Evolution of Robots. ● Evolution of Artificial Intelligence. **Procedural Knowledge:** ● Understand and describe the evolution of the internet and the world wide web. ● Explain how communication takes place over the internet and how information is found on the world wide web.	**Composite Knowledge:** The Year 7 and 8 Programming units are prerequisites for this unit. Whilst the first two lessons do reconnect students with Python and makes sure they can remember how to read and create simple programs using inputs, outputs and selection, it is assumed that learners are already able to write Python programs that display messages, receive keyboard input, use simple arithmetic expressions, and control the flow of program execution through selection. This unit will then introduce the concept of String manipulation methods, List data structures and List operations. Students will adopt the five stages of PRIMM within their program development activities. **Component Knowledge:** **Foundational Knowledge:** Declarative Knowledge: ● Sequence / Selection / Iteration. ● String Manipulation Methods. ● List Data Structures. ● List Operations. **Procedural Knowledge:** ● Write programs that display messages, receive keyboard input, and use simple arithmetic expressions in assignment statements. ● Use selection statements to control the flow of program execution. ● Create lists.

● Recognise how human errors pose security risks to data. ● Define hacking in the context of cyber security. ● Explain how a DDoS attack can impact users of online services. ● Identify strategies to reduce the chance of a brute force attack being successful. ● Explain the need for the Computer Misuse Act. ● List the common malware threats. ● Examine how different types of malware cause problems for computer systems. ● Understand encryption and encrypt and decrypt secret messages using encryption keys. **<u>Upper Hierarchical Knowledge</u>** ● Explain how malicious bots can have an impact on societal issues. ● Identify the most effective methods of preventing cyber-attacks.	● Understand how mobile phones were invented, how they work and the difference between the mobile phone generations. ● Understand how satellites have evolved from the beginnings of the space race to today, and how they communicate with earth. ● Understand how robots have evolved from science fiction to reality. ● Understand the basics of AI, what it can and currently cannot do. **<u>Upper Hierarchical Knowledge</u>** ● Describe the impact the internet and the WWW has had on the world. ● Describe the impact mobile phones are having on the world. ● Describe the impact satellites are having on the world. ● Describe the impact robots are having on the world. ● Describe the impact AI is having on the world.	● Access individual list items. ● Perform common operations on lists or individual items. ● Perform common operations on strings or individual characters. **<u>Upper Hierarchical Knowledge</u>** ● Combine key programming language features to develop solutions to meaningful problems. ● Independently locate and correct common syntax errors.
Schemata 4: Python Programming 4	**Schemata 5: 2D Animation**	**Schemata 6: HTML / CSS**
<u>Composite Knowledge:</u> This unit is a continuation of the previous unit (Python Programming 3) where students will continue learning about List data structures, more specifically how to navigate those data structures using both while and for loops. Students will also learn about data validation, authentication and error handling before embarking on a mini project which will allow them to apply the knowledge and skills they have acquired so far. **<u>Component Knowledge:</u>** **Foundational Knowledge:** **Declarative Knowledge:** ● While Loops. ● For Loops. ● Data Validation. ● Authentication.	**<u>Composite Knowledge:</u>** In this unit learners will discover how to create 2D animations using the on-line tool Wick Editor. By completing this unit learners will gain a greater understanding of how this important creative field of animation is used to make the media products that we consume. Sessions will take learners through the basics of frame-by-frame animation, “Tweening”, layers, creating assets and animating a robot. Students will then be able to create an animation of their choice and export it to a GIF file. **<u>Component Knowledge:</u>** **Foundational Knowledge:** Declarative Knowledge: ● Animation Types. ● Frames / Onion skinning / Layers. ● Tweening. ● Assets.	**<u>Composite Knowledge:</u>** In this unit, learners will explore the building blocks of the World Wide Web, by learning to use HTML tags, and CSS code. Students will start to understand how web pages are constructed using HTML tags, and how they can be modified to start to resemble the websites they are accustomed to, including bringing them to life through the use of images. By the end of the unit, learners will have a functioning website. **<u>Component Knowledge:</u>** **Foundational Knowledge:** Declarative Knowledge: ● HTML Tags. ● CSS Code. ● Images in Web Pages. ● Hyperlinks between Web Pages.

● Error Handling. **Procedural Knowledge:** ● Use iteration (while statements) to control the flow of program execution. ● Use iteration (for statements) to iterate over list items. ● Use variables to keep track of counts and sums. ● Define what validation means. ● Write simple validation programs. ● Define what authentication means. ● Write simple authentication programs. ● Understand how to handle errors in Python programs. **Upper Hierarchical Knowledge** ● Combine key programming language features to develop solutions to meaningful problems. ● Independently locate and correct common syntax errors.	● Export to GIF files. **Procedural Knowledge:** ● Understand the different types of animation. ● Create frame by frame animations using animation software. ● Use of layers when creating animations. ● Understand the purpose of Tweening and why it's used in animation. ● Create animations using Tweening. ● Create own assets in Wick Editor by creating a traffic light animation. ● Identify a range of tools used in animation software. **Upper Hierarchical Knowledge** ● Create a robot animation that successfully assembles itself together. ● Add flashing elements to the robot animation. ● Enhance the movement of the animation by making the robot wave.	**Procedural Knowledge:** ● Describe what HTML is. ● Use HTML to structure and modify web pages. ● Display images within a web page. ● Describe what CSS is. ● Use CSS to style statice web pages ● Describe what a hyperlink is. ● Create hyperlinks to allow users to navigate between multiple web pages. **Upper Hierarchical Knowledge** ● Create own functioning website.

Year 9 Final Composite Knowledge End Point

By the end of year 9 the Computer Science department aims to ensure all students have a deep understanding of the threats to computer systems, computer networks and our data as well as the impact technology has had on our world, enabling them to understand the importance of using the technology safely, respectfully, responsibly and securely. Building on their year 8 skills, they will have undertaken additional creative projects to achieve challenging goals, including solving a variety of real-world computational problems through computational thinking and programming activities using data structures in a text-based programming language. This lays the final foundation for all programming work at KS4. Finally, students will gain skills and knowledge in creating both a 2-D animation and their own websites, further widening their experience in undertaking creative projects.

ARU.